



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/780,452	06/18/2001	Vladimir I. Miloushev	99300/0274763	1119
7590	08/24/2005			
EXAMINER				FRANCIS, MARK P
ART UNIT				PAPER NUMBER
				2193

DATE MAILED: 08/24/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/780,452	MILOUSHEV ET AL.
	Examiner	Art Unit
	Mark P. Francis	2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 18 June 2001.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-133 is/are pending in the application.
 - 4a) Of the above claim(s) 49-132 is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-48, and 133 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 18 June 2001 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date: _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date: _____ | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the application filed on June 18, 2001.
2. Claims 1-48 and 133 have been examined.

Oath/Declaration

3. The Office acknowledges receipt of a properly signed oath/declaration filed June 18, 2001.

Priority Date

4. The priority date considered for this application is August 16, 1999.

Claim Objections

5. Claims 29 and 38 are objected to under 37 CFR 1.75(c) as being in improper form because a multiple dependent claim should refer to other claims in the alternative only. See MPEP § 608.01(n). Accordingly, claims 29 and 38 have not been further treated on the merits.

For examination purposes, claim 29 will depend upon either claim 25 or claim 28 and claim 38 will depend upon either claim 34 or claim 36.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claims 1-48 and 133 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

With respect to claims 1,9,10,11,13-15,23,26,31,34,36,39,41,43,47, and 133,

Art Unit: 2193

Statutory subject matter requires two things:

- (1) it must be in the "useful arts," U.S. Const., art. I, 8, cl.8,

In this instance, the language of the claims raises a question as to whether the claims are directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a useful, concrete, and tangible result to form the basis of statutory subject matter under 35 U.S.C. 101; and if it is,

- (2) it must not fall within one of the exceptions for "laws of nature, physical phenomena and abstract ideas."

Under the most recent Federal Circuit cases, transformation of data by a machine (e.g., computer) is statutory subject matter provided the claims recite a "practical application, which produce[s] a useful, concrete and tangible result."

State St. Bank & Trust Co. v. Signature Fin. Group, Inc. 149 F.3d 1368, 1373, 47

USPQ2d 1596, 1600-01 (Fed. Cir. 1998).

In this instance, the language of the claim raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a useful, concrete and tangible result to form the basis of statutory subject matter under 35 USC 101.

Furthermore, the Office's interpretation of this claim is that it does not expressly or implicitly require performance of any of the steps by a machine such as general-purpose digital computer. Structure will not be read into the claims for the purpose of

the statutory subject matter analysis even though the steps might be capable of being performed by a machine.

Applicants fail to disclose that the elements of the software system are tangibly embodied in and executed by a piece of hardware and that their functions have practical applications which produce useful, concrete and tangible results under the State Street Formulation.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

9. A person shall be entitled to a patent unless --

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1-22, 34-38, and 133 are rejected under 35 U.S.C. 102(e) as being anticipated by Kimura. (U.S. Pat 6,304,922)

Independent claims

With respect to claims 1 and 133, Kimura discloses A method for designing a software system in which system(Col 1:65-67, Col 2:1-13, "...providing a method and system...") at least a first object is created arbitrarily earlier than a second object(Col 2:23-46, "...at

Art Unit: 2193

least one second control object is created by a first control object...”, Col 4:1-40, “...and common objects OCX...”, e.g. See Fig. 1 and related text) and said second object is automatically connected to at least said first object, said method comprising the steps of: creating said first object; (Col 2:23-46, “...at least one second control object is created by a first control object...”, Col 4:1-40, “...and common objects OCX...”, e.g. See Fig. 1 and related text)

creating a first container object capable of holding at least one other object of arbitrary object class; (Col 2:23-46, “...at least one second control object is created by a first control object...”, Col 4:1-40, “...and common objects OCX...”, Col 4:9-50, “...the container application...”, e.g. See Fig. 1 and related text)

defining at least a first template connection between said first object and said first container object; (Col 4:29-50, “...the connection point...”, Col 6:52-67, “...setting connection...”, Col 7:30-57, “...to IConnectionPoint,...”, Col 8:35-67, “...generates plural objects...”, Col 14: 36-67, “...a connection between IConnectionPoint...”, e.g. See Figs. 1,5,6, and 7)

creating said second object; (Col 2:23-46, “...at least one second control object is created by a first control object...”, Col 4:1-40, “...and common objects OCX...”, e.g. See Fig. 1 and related text)

connecting said second object to said first object using said first template connection in which template said first container object is replaced with said second object. (Col 4:29-50, “...the connection point...”, Col 6:52-67, “...setting connection...”, Col 7:30-57, “...to

IConnectionPoint,...”, Col 8:35-67, “...generates plural objects...”, Col 14: 36-67, “...a connection between IConnectionPoint...”, e.g. See Figs. 1,5,6, and 7)

With respect to claims 9 and 10, Kimura discloses a method for describing connections between a first plurality of objects in a software system and a second plurality of objects in said software system(Col 1:65-67,Col 2:1-13, “...providing a method and system...”), said second plurality being created arbitrarily later than said first plurality; (Col 2:23-46, “...at least one second control object is created by a first control object...”,Col 4:1-40, “...and common objects OCX...”,e.g. See Fig. 1 and related text), said method comprising the steps of:

defining at least a first object of said first plurality; (Col 2:23-46, “...at least one second control object is created by a first control object...”,Col 4:1-40, “...and common objects OCX...”,e.g. See Fig. 1 and related text)

defining a first container object which will be used as a placeholder for defining connections between said first object and each object of said second plurality; (Col 2:23-46, “...at least one second control object is created by a first control object...”,Col 4:1-40, “...and common objects OCX...”, Col 4:9-50, “...the container application...”,e.g. See Fig. 1 and related text)

defining at least a first connection to be created between said first object and each object of said second plurality as a connection between said first object and said first container object. (Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint...", Col 8:35-67, "...generates plural objects...", Col 14: 36-67, "...a connection between IConnectionPoint...", e.g. See Figs. 1,5,6, and 7)

With respect to claim 11, Kimura discloses In a software system, said software system having a plurality of objects, a container object comprising:
a first memory for keeping reference to at least a first object of arbitrary object class; (Col 4:40-67, "...memory is reserved for the stack...", Col 5:1-19, "...memory areas...")
a section of program code causing said first memory to be modified so that it will contain a first reference to a second object; Col 7:30-67, "...a connection for events from second OCX 11 is set...")
a section of program code accessing a data structure(Col 4:9-39, "...data storage structures...", Col 7:30-67, "...the array M_dispID is defined...") and determining that at least a first connection needs to be established between said second object and at least a third object;(Col 5:34-67, "...provides an interface object...", Col 6:52-67, "...process fir setting the connection between second OCX 11 and interface object...", Col 7: 1-67, "...a connection for events from second OCX 11 is established...")

Art Unit: 2193

a section of program code causing said first connection to be established.(Col 6:52-67,
“...setting the connection...”, Col 7:30-57, “...a connection for events...”, Col 14:48-67,
“...a connection between IconnectionPoint... ”, e.g. See Fig. 1,5, and 7, and related text)

With respect to claim 13, Kimura discloses In a software system, said software system having a plurality of objects, a container object comprising:

a memory for keeping at least one reference to a contained object of arbitrary class;
(Col 4:40-67, “...memory is reserved for the stack...”, Col 5:1-42, “...memory areas...”)
a section of program code causing said first memory to be modified so that it will contain a first reference to a second object; Col 7:30-67, “...a connection for events from second OCX 11 is set...”)

a connection point for receiving requests to modify the set of contained objects; (Col 6:52-67, “...setting the connection...”, Col 7:30-57, “...a connection for events...”, Col 14:48-67, “...a connection between IConnectionPoint... ”, e.g. See Fig. 1,5, and 7, and related text)

at least one virtual connection point that accepts at least a first connection to be established to said contained object, said acceptance occurring before said contained object is added to said contained object; (Col 4:9-50, “...If the connection point for the event does not have a valid interface...”)

Art Unit: 2193

a section of program code that establishes said first connection when said contained object is added to said container object.(Col 6:44-67, "...setting the connection...", Col 7:35-61, "...a connection fir events...", e.g. See Figs. 2 and 3 and related text)

With respect to claim 14, Kimura discloses In a software system, said software system having a plurality of objects, a container object comprising:

a first memory for keeping at least one reference to a contained object of arbitrary class; (Col 4:40-67, "...memory is reserved for the stack...", Col 5:1-42, "...memory areas...")
a section of program code causing said first memory to be modified so that it will contain a first reference to a second object; Col 7:30-67, "...a connection for events from second OCX 11 is set...")

a connection point for receiving requests to modify the set of contained objects; (Col 6:52-67, "...setting the connection...", Col 7:30-57, "...a connection for events...", Col 14:48-67, "...a connection between IConnectionPoint...", e.g. See Fig. 1,5, and 7, and related text)

at least one virtual property that accepts the value to be set in a first property on said contained object, said virtual property being capable of accepting values of a plurality of data types; (Col 4:9-50, "...If the connection point for the event does not have a valid interface...", Col 8:1-67, "...if the properties,methods,...")

a section of program code that sets said first property on said contained object to said accepted value when said contained object is added to said contained object.(Col 6:44-

67, "...setting the connection...", Col 7:35-61, "...a connection fir events...", e.g. See Figs. 2 and 3 and related text)

With respect to claim 15, Kimura discloses In a software system, said software system having a plurality of objects, a container object comprising:

a first memory for keeping a first plurality of contained objects of arbitrary classes; (Col 4:40-67, "...memory is reserved for the stack...", Col 5:1-42, "...memory areas...")

a second memory for keeping a second plurality of unique identifiers, each identifier of said second plurality associated with exactly one object of said first plurality; (Col 7:30-67, "...the event dispatch ID...", Col 14:24-62, "...passes an address identifier...")

at least a first property, said first property being a second property of a first object of said first plurality and said first property being identified by a combined identifier produced by combining the associated identifier of said first object and the identifier of said second property. (Col 7:30-67, "...the event dispatch ID...", Col 14:24-62, "...passes an address identifier...")

With respect to claims 18 and 21, Kimura discloses In a software system, said software system having a plurality of objects, each object of said plurality of objects belonging to an object class, said software system having means for building at least one structure of connected objects and means of describing said structure of connected objects, a container object class comprising:

means for holding a plurality of contained objects, said means being applicable to contained objects of any class; (Col 2:23-46, "...at least one second control object is created by a first control object...", Col 4:1-40, "...and common objects OCX...", Col 4:9-50, "...the container application...", e.g. See Fig. 1 and related text)

means for changing the set of said contained objects programmatically, said means being applicable to contained objects of any class; (Col 2:32-56, "...execute processes appropriate...")

means for presenting said plurality of contained objects as a single object in said structure of connected objects, said means being applicable to contained objects of any class. (Col 2:47-67, "...a first control object thus comprises a second interface...")

With respect to claim 34, Kimura discloses A method for caching and propagating property values to a dynamic set of objects in a software system(e.g. See Figs. 1, 5, and 7 and related text), said software system having a plurality of objects(Col 2:15-46, "...plural control objects..."), each of said objects having a plurality of properties(Col 8:1-34, "...if the properties...", Col 14:48-67, "...for supplying the properties and events..."), each said property having a value and an identifier(Col 16:1-67, "...for passing the identifier...the current event type as a parameter is passed..."), said method comprising the steps of: accepting a first request to modify the value of a first property on behalf of said dynamic set of objects as if said dynamic set of objects were one object; Col 6:52-67, "...When EstablishConnection is called...", Col 14:48-67, "...for supplying the properties...")

storing said value and identifier of said first property in a first data storage(Col 8:8-67, "...If the parameters are valid...a unique ID enabling..."); retrieving said value and identifier of said first property from said first data storage; issuing a request to modify the value of said first property on a first object of said dynamic set of objects, using said value and identifier retrieved from said first data storage. (Col 8:8-67, "...If the parameters are valid, the dispIDMember array of events obtained...a unique ID enabling...")

With respect to claim 35, the rejection of claim 34 is incorporated and further, Kimura discloses A container object in a software system.(Col 4: 40-67, "...Container application...")

With respect to claim 36, Kimura discloses a method for caching and propagating outgoing connections of a dynamic set of objects in a software system(e.g. See Figs. 1, 5, and 7 and related text), said software system having a plurality of objects(Col 2:15-46,"...plural control objects..."), said software system having means for establishing connections between said objects(Col 6:44-67, "...When EstablishConnection is called..."), said connections providing means for a first connected object to make outgoing calls to a second connected object(Col 7:30-67,"...a connection for events from second OCX 11..."), said method comprising the steps of: accepting the request to establish a first outgoing connection between said dynamic set of objects and a first object, as if said dynamic set of objects were a single object; (Col

6:52-67, "...When EstablishConnection is called..."), storing a first data value necessary to effect said first connection in a first data storage; retrieving said first data value from said first data storage; (Col 7:1-67, "...and the parameter data list is stored to ParamInfo...")

issuing a request to establish a second connection between a second object of said dynamic set and said first object, using said first data value retrieved from said first data storage. (Col 8:35-67, "...second OCX with first OCX...")

11. Claims 26-28 are rejected under 35 U.S.C. 102(e) as being anticipated by Brobst. (U.S. Pat 6,427,229)

With respect to claim 26, Brobst discloses A container object(Col 6:56-67, "...The container class...") in a software system(e.g. See Fig. 1 and related text), said software system having at least one first object and said container object(Col 6:56-67, "...The container class...",e.g. See Fig. 1 and related text), said first object having at least one first property(Col 4:51-67, "...can hold a collection of object oriented objects..."), said software system having means of requesting operations over said first property, said container comprising:

means for adding and removing said first object from said container;(Col 4:51-67, "...for adding, removing, and replacing...", Col 6:26-55, "...the logic for adding, removing, and replacing elements...", Col 8:1-21, "...which defines add(),remove(),....)

means for defining a second property on said container object;(Col 7:36-55, "...subclass of a PropertyContainer class...", Col 8:61-67, and Col 9:1-30, "...PropertyContainer class...", Col 9:46-67, "...addPropertyBy(),..., e.g. See Fig. 4, element 310, and related text)

means for transforming a request for operations over said second property into a request for operations over said first property. (Col 3:48-67 and Col 4:1-22, "when requested by client...request messages...", Col 9:11-67, "...This operation...is a request...", e.g. See Figs. 4 and 5 and related text)

Dependent claims

With respect to claim 2, the rejection of claims 1 is incorporated respectively and further, Kimura discloses wherein the step of creating said second object is performed by said first container object. (Col 4:13-50, "...container application creates...", e.g. See Figs. 1 and 5)

With respect to claim 3, the rejection of claim 1 is incorporated and further, Kimura discloses wherein the step of connecting said second object to said first object is performed by said first container object. (Col 4:13-50, "...container application creates...", Col 6:52-67, "...setting the connection...", Col 7:30-57, "...the event connection...",e.g. See Figs. 1 and 5 and related text)

Art Unit: 2193

With respect to claim 4, the rejection of claim 1 is incorporated and further, Kimura discloses wherein the step of creating said second object is performed by said first container object and the step of connecting said second object to said first object is performed by said first container object. (Col 4:13-50, "...container application creates...", Col 6:52-67, "...setting the connection...", Col 7:30-57, "...the event connection...", e.g. See Figs. 1 and 5 and related text)

With respect to claim 5, the rejection of claim 1 is incorporated and further, Kimura discloses connections between all objects are established between connection points on said objects. (Col 4:13-50, "...container application creates...", Col 6:52-67, "...setting the connection...", Col 7:30-57, "...the event connection...", e.g. See Figs. 1,5, and 7 and related text)

With respect to claim 6, the rejection of claim 1 is incorporated and further, Kimura discloses said first template connection is defined in a data structure. (Col 7:20-67, "...the event array of the common object...between the arrays when establishing the event connection....")

With respect to claim 7, the rejection of claim 5 is incorporated and further, Kimura discloses wherein said first template connection is defined in a data structure. (Col 7:20-67, "...the event array of the common object...between the arrays when establishing the event connection....")

With respect to claim 8, the rejection of claim 2 incorporated and further, Kimura discloses A system created using any one of claims 1, 2, 3, 4, 5, 6 or 7. (Col 4:13-50, "...container application creates...", e.g. See Figs. 1 and 5)

With respect to claim 12, the rejection of claim 10 is incorporated and further, Kimura discloses a section of program code causing said second object to be created. ;(Col 2:23-46, "...at least one second control object is created by a first control object...", Col 4:1-40, "...and common objects OCX...",e.g. See Fig. 1 and related text)

With respect to claim 16, the rejection of claim 15 is incorporated and further, Kimura discloses wherein each said property comprises a terminal. (Col 9:56-67, "...built around a personal computer...", Col 12:7-43, "...One property of printer driver...",e.g. See Fig. 8 and related text)

With respect to claim 17, the rejection of claim15 is incorporated and further, Kimura discloses wherein the second memory doesn't exist and contained objects are identified by identifiers assigned by the container. (Col 4:40-67, "...Container application...memory is reserved...", Col 5:1-42, "...other requisite memory areas...",Col 8:35-67, "...a unique ID...")

With respect to claim 19, the rejection of claim 18 is incorporated and further, Kimura

Art Unit: 2193

discloses wherein said single object is an instance of said container object class. (e.g.
See Figs. 1,5, and 7 and related text)

With respect to claim 20, the rejection of claim 18 is incorporated and further, Kimura discloses a container object which is an instance of the container object class.(e.g. See Figs. 1,5, and 7 and related text)

With respect to claim 22, the rejection of claim 21 is incorporated and further, Kimura discloses a container object which is an instance of the container object class. (e.g. See Figs. 1,5, and 7 and related text)

With respect to claim 27, the rejection of claim 26 is incorporated and further, Brobst discloses wherein said software system has means of identifying said first property using a first identifier, said container object having the additional means to identify said second property using said first identifier. (Col 7:36-67, "...which provides a unique handle identifier instance that is associated with each key instance...", Col 8:4-67, "...is an identifier within an object oriented...")

With respect to claim 28, the rejection of claim 26 is incorporated and further Brobst discloses wherein said software system has means of identifying said first property using a first identifier, said container object having the additional means to identify said first object using a second identifier and said container object having the additional

Art Unit: 2193

means to identify said second property using a combination of said first identifier and said second identifier. (Col 7:20-67, "...which provides a unique handle identifier instance that is associated with each key instance...", Col 8:4-67, "...is an identifier within an object oriented...")

With respect to claim 29, the rejection of claim 28 is incorporated and further, Brobst discloses A container object having the sum of the means of the container object of claim 28. (Col 4:51-67, "...container class defines an object...", Col 6:56-67, "...The container class...", Col 7:21-67, "...the container class...")

With respect to claim 30, the rejection of claim 29 is incorporated and further, Brobst discloses wherein all the specified means of said container are implemented independently of the class of said first object. (Col 4:51-67, "...container class defines an object...", Col 6:56-67, "...The container class...", Col 7:21-67, "...the container class...")

With respect to claim 33, the rejection of claim 31 is incorporated and further, Brobst discloses wherein said unique identifier is assigned outside of said container, and said container has the additional means of associating said unique identifier with each said contained object. (Col 7:21-67, "...the container class...", e.g. See Fig. 3, elements 320 Handle and related text)

Art Unit: 2193

With respect to claim 37, the rejection of claim 36 is incorporated and further, Kimura discloses A container object in a software system.(Col 4:40-67, "...Container application...", Col 5:20-42, "...a specific container application...")

With respect to claim 38, the rejection of claim 36 is incorporated and further, Kimura discloses A container object in a software system. (Col 4:40-67, "...Container application...", Col 5:20-42, "...a specific container application...")

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claims 23-25, 31-33,39-42, and 43-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kimura in view of Brobst. (6,427,229)

Regarding claim 23,

Kimura discloses In a software system having at least a first object and a second object, said first object having at least one first connection point(Col 2:23-46, "...at least one second control object is created by a first control object...",Col 4:1-40, "...and common objects OCX...",e.g. See Fig. 1 and related text), said second object having at least one

Art Unit: 2193

second connection point, said first connection point being used to establish a first connection between said first connection point of said first object and said second connection point of said second object, (e.g. See Figs. 1,5,6, and 7 and related text)

and said software system having means of requesting the establishment of a connection between connection points, a container object(Col 2:23-46, "...at least one second control object is created by a first control object...", Col 4:1-40, "...and common objects OCX...", Col 4:9-50, "...the container application...", e.g. See Fig. 1 and related text) comprising:

means for defining a third connection point on said container object; (Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", Col 8:35-67, "...generates plural objects...", Col 14: 36-67, "...a connection between IConnectionPoint...", e.g. See Figs. 1,5,6, and 7)

means for transforming a requests for establishing of a connection between said second connection point and said third connection point into a request for establishing a connection between said second connection point and said first connection point(Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", Col 8:35-67, "...generates plural objects...", Col 14: 36-67, "...a connection between IConnectionPoint...", e.g. See Figs. 1,5,6, and 7)

Art Unit: 2193

but does not disclose means for adding and removing said first object from said container;

Brobst discloses means for adding and removing said first object from said container(Col 4:40-67, "...The container class...logic for adding, removing, and replacing data...", Col 6:26-67, "...the logic for adding, removing, and replacing...", Col 7:21-35, "...to add, remove, or replace...", Col 8:1-21, "...which defines add(),...") in an analogous system for the purpose of providing an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst:Col 4:12-23)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include means for adding and removing objects from a container class.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst: Col 4:12-23)

Regarding claim 24,

Art Unit: 2193

wherein said software system includes means of identifying said first connection point using a first identifier, said container object having the additional means to identify said third connection point using said first identifier. (Kimura: Col 6:52-67, "...When EstablishConnection is called...", Col 8:35-67, "...a unique ID...", Col 14:24-67, "...passes an address identifier and opens a connection...", e.g. See figs 1 and 5 and related text)

Regarding claim 25,

wherein said software system includes means of identifying said first connection point using a first identifier, said container object having the additional means to identify said first object using a second identifier and said container object having the additional means to identify said third connection point using a combination of said first identifier and said second identifier. (Kimura: Col 6:52-67, "...When EstablishConnection is called...", Col 8:35-67, "...a unique ID...", Col 14:24-67, "...passes an address identifier and opens a connection...", e.g. See Fig. 5 and related text)

Regarding claim 31,

Kimura discloses A container object(Col 2:23-46, "...at least...", Col 4:9-50, "...the container application..." e.g. See Fig. 1 and related text) in a software system, said software system having a plurality of objects(Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", e.g. See

Figs. 1,5,6, and 7 and related text), said software system having means for requesting operations over an object, said container object comprising:

means for holding a plurality of contained objects; (Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", e.g. See Figs. 1,5,6, and 7 and related text)

means for identifying each object of said contained objects by a separate, unique identifier for each object; (Col 6:52-67, "...(the variable storing a 128-bit value identifying...)...", Col 8:35-67, "...a unique ID enabling each interface object...")

means of handling requests for operations over any object of said contained objects wherein said identifier is used to determine which object of said contained objects should handle the request;(Col 8:45-67, "...The first OCX can therefore clearly determine...by referencing the unique ID...") but does not disclose means for changing the set of said contained objects programmatically;

Brobst discloses means for changing the set of said contained objects programmatically; (Col 4:40-67, "...The container class...logic for adding, removing, and replacing data...", Col 6:26-67, "...the logic for adding, removing, and replacing...", Col 7:21-35, "...to add, remove, or replace...", Col 8:1-21, "...which defines add(),...") in an analogous system for the purpose of providing an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst:Col 4:12-23)

Art Unit: 2193

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include means for modifying the set of contained objects.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst: Col 4:12-23)

Regarding claim 32, Brobst discloses wherein said container has the additional means of automatically assigning said unique identifier to each object added to said container. (Col 7:35-67, "...which provides a unique identifier...")

Regarding claim 33, Brobst discloses wherein said unique identifier is assigned outside of said container, and said container has the additional means of associating said unique identifier with each said contained object. (Col 7:35-67, "...which provides a unique identifier...", Col 8:22-67, "...is an identifier...")

Regarding claims 39 and 41,

Kimura discloses A container object(Col 2:23-46, "..at least...", Col 4:9-50, "...the container application..." e.g. See Fig. 1 and related text) in a software system, said software system having a plurality of objects(Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", e.g. See

Figs. 1,5,6, and 7 and related text), said software system having means of building at least one structure of connected objects, said software system having a first means of describing said structure, said software system providing a second means of enumerating all connections described by said first means, said container being a first object in said structure(e.g. See Fig. 1,5, and 7, element 20, and related text), said container being connected to at least a second object in said structure, (e.g. See Fig. 1,5, and 7, element 20, and related text) said container comprising:

means for holding a plurality of contained objects; Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", e.g. See Figs. 1,5,6, and 7 and related text)

means for finding a first described connection between said container and said second object; Col 4:29-50, "...the connection point...", Col 6:52-67, "...setting connection...", Col 7:30-57, "...to IConnectionPoint,...", e.g. See Figs. 1,5,6, and 7 and related text)

means for establishing said first connection between a third object contained in said container and said second object(Col 6:35-67, "...second OCX objects 11a and 11b...",e.g. See Fig. 5, elements 10,11a,11b,20,15b and related text) but does not disclose means for changing the set of said contained objects programmatically;

Brobst discloses means for changing the set of said contained objects programmatically; (Col 4:40-67, "...The container class...logic for adding, removing, and replacing data...",Col 6:26-67, "...the logic for adding, removing, and replacing...", Col 7:21-35, "...to add, remove, or replace...", Col 8:1-21, "...which defines add(),...") in an

Art Unit: 2193

analogous system for the purpose of providing an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst:Col 4:12-23)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include means for modifying objects from a container class.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst: Col 4:12-23)

Regarding claim 40,

Kimura discloses wherein said container has the additional means of establishing all connections between said container and other objects in said structure, said all connections being described by said first means, said additional means causing the establishing of each of said all connections between each of said contained objects and said other objects in said structure. (Kimura: e.g. See Fig. 1,5,6 and 7, and related text)

Regarding claim 42,

Kimura discloses wherein said container establishes connections between a first connection point of said third object and a second connection point of said second

object. (Kimura: e.g. See Fig. 1,5,6 and 7, and related text)

Regarding claims 43 and 47,

Kimura discloses A container object(Col 2:23-46, “..at least...”, Col 4:9-50, “...the container application...” e.g. See Fig. 1 and related text) in a software system(Col 1:65-67,Col 2:1-13, “...providing a method and system...”), said software system having a plurality of objects, said container having a first connection to at least one object(Col 4:29-50, “...the connection point...”, Col 6:52-67, “...setting connection...”, Col 7:30-57, “...to IConnectionPoint,...”, e.g. See Figs. 1,5,6, and 7 and related text), said first connection being described in a first data structure, said container comprising: means for holding a plurality of contained objects; (Col 2:16-46, “...plural control objects...”)

means for establishing said first set of connections. (Col 6:44-67, “...When EstablishConnection is called,...”) but does not disclose means for changing the set of said contained objects programmatically;

means for determining a first set of connections to be established for each object added to said set of contained objects based on the set of connections described in said first data structure;

Brobst discloses means for changing the set of said contained objects

programmatically; means for determining a first set of connections to be established for each object added to said set of contained objects based on the set of connections described in said first data structure; (Col 4:40-67, "...The container class...logic for adding, removing, and replacing data...", Col 6:26-67, "...the logic for adding, removing, and replacing...", Col 7:21-35, "...to add, remove, or replace...", Col 8:1-21, "...which defines add(),..."") in an analogous system for the purpose of providing an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst:Col 4:12-23)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include means for adding and removing objects from a container class.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide an object oriented mechanism and method to allow a user to define maintenance constraints and a simple maintenance interface on a collection.(Brobst: Col 4:12-23)

Regarding claim 44, Kimura discloses wherein said container further comprises means for dissolving said first set of connections. (e.g. See Figs. 1,5, and 7 and related text)

Regarding claim 45, Kimura discloses wherein said container further comprises: means

Art Unit: 2193

for remembering a second set of outgoing connections from said container to other objects means for excluding said second set of connections from said first set of connections means for establishing said second set of outgoing connections for each object added to said set of contained objects. (e.g. See Figs. 1,5, and 7 and related text)

Regarding claim 46, Kimura discloses wherein said container further comprises: means for remembering properties set on said container; means for setting remembered properties on each new object added to said set of contained objects; means for propagating properties set on said container to all objects in said set of contained objects; (Col 8:1-35, "...if the properties...", Col 12:6-32, "...One property...", Col 14, 48-67, "...for supplying the properties...", e.g. See Figs. 1,5, and 7 and related text)

Regarding claim 48, Kimura discloses wherein said container further comprises: means for remembering properties set on said container; (Col 8:1-35, "...if the properties...", Col 12:6-32, "...One property...", Col 14, 48-67, "...for supplying the properties...", e.g. See Figs. 1,5, and 7 and related text)

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mark P. Francis whose telephone number is (571) 272-7956. The examiner can normally be reached on Mon-Fri 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mark P. Francis

Patent Examiner

Art Unit 2193

Kakali Chan
KAKALI CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100